

```

*****
; PROGRAM ID:      SYSTEM BOOTSTRAP DRIVER
*****
; PROPERTY OF:      JADE COMPUTER PRODUCTS
;                   4901 W. ROSECRANS BLVD.
;                   HAWTHORNE, CALIFORNIA
;                   90250, U.S.A.
*****
; VERSION:          2.2
*****
; THE SYSTEM BOOTSTRAP DRIVER IS ONE OF TWO MODULES
; THAT MAKE UP THE SYSTEM RESIDENT BOOTSTRAP. THIS
; MODULE IS TO BE EXECUTED BY THE SYSTEM PROCESSOR.
; DURING EXECUTION, THIS MODULE PERFORMS A BLOCK MOVE
; OF THE SECOND MODULE (BOOT INJECTION MODULE) INTO
; THE DOUBLE D CONTROLLER MEMORY. A SUCCESSFUL BOOT
; OPERATION BY THE DOUBLE D WILL LEAVE DCM IN BANK 0
; AND BIOS IN BANK 1. THE REMAINDER OF THIS MODULE
; THEN MOVES THE BIOS IMAGE TO THE PROPER SYSTEM
; ADDRESS AND JUMPS TO THE BIOS COLD START ENTRY.
***** SK ***
;
; DOUBLE D HARDWARE PARAMETERS. PLEASE NOTE THIS
; SECTION CONTAINS CONDITIONAL STATEMENTS.
*****
;
0043 = D$PORT EQU 043H ;DOUBLE D PORT ADDRESS.
0001 = TRUE EQU 1 ;TRUE IS A ONE.
0000 = FALSE EQU 0 ;FALSE IS A ZERO.
0001 = REV$B EQU TRUE ;SET TRUE FOR REV B BOARDS.
0000 = REV$C EQU FALSE ;SET TRUE FOR REV C BOARDS.
0000 = MA10 EQU FALSE ;TRUE IF MA10 JUMPED (REV-B).
;
        IF REV$B
0002 = DS$HLT EQU 002H ;STATUS PORT HALT INDICATOR.
000C = DS$ASW EQU 00CH ;STATUS PORT ADDR SW MASK.
D$BASE SET 0E400H ;SYSTEM WINDOW BASE ADDRESS
ENDIF
;
        IF MA10
D$BASE EQU 0E000H ;SYSTEM WINDOW BASE ADDRESS
ENDIFF
;
        IF REV$C
DS$HLT EQU 001H ;STATUS PORT HALT INDICATOR.
DS$ASW EQU 00EH ;STATUS PORT ADDR SW MASK.
D$BASE EQU 0E000H ;SYSTEM WINDOW BASE ADDRESS
ENDIF
;
***** BOOTSTRAP INJECTION MODULE PARAMETERS *****
;
1000 = IM$ADR EQU 1000H ;BOOT INJECTION MODULE ADDRESS.
00C0 = IM$SIZE EQU 0000H ;BOOT INJECTION MODULE SIZE.
;
*****

```

```

; BOOTSTRAP LINKAGE ADDRESS. *
; ****
;
0080 = BSTACK EQU 0080H ; BOOTSTRAP TOP OF STACK.
0040 = D$ADDR EQU 0040H ; DOUBLE D ADDRESS POINTER.
0377 = BL$DCS EQU 0377H ; DCM DISK CONTROLLER STATUS.
0378 = BL$ADR EQU 0378H ; DCM LOAD AND JUMP ADDRESS.
037A = BL$BSZ EQU 037AH ; DCM BLOCK LOAD SIZE.
;
; ****
; DOUBLE D HARDWARE COMMANDS *
; ****
;
0080 = DC$BGN EQU 080H ; RESET Z80A AND EXECUTE.
0001 = DC$MRQ EQU 001H ; REQUEST MEMORY WINDOW.
0000 = DC$MRT EQU 000H ; RELEASE MEMORY WINDOW$.
0001 = DC$MBO EQU 001H ; SELECT MEMORY BANK 0.
0003 = DC$MB1 EQU 003H ; SELECT MEMORY BANK 1.
0002 = DC$EXC EQU 002H ; ISSUE DOUBLE D INTERRUPT.
;
; ****
; ASSEMBLER DIRECTIVES *
; ****
;
0100 ORG 0100H ; MODULE ADDRESS (ALTERABLE).
;
; ****
; SET STACK AND CONTROLLER ADDRESS *
; ****
;
0100 318000 BEGIN: LXI SP,BSTACK ; SET STACK POINTER.
0103 DB43 IN D$PORT ; INPUT STATUS PORT.
0105 E60C ANI DS$ASW ; MASK FOR ADDR SWS.
0107 07 RLC ; POSITION BITS.
0108 F6E4 ORI D$BASE SHR 8 ; OR IN BASE ADDR.
010A 67 MOV H,A ; HIGH BYTE VALUE.
010B 2E00 MVI L,0 ; LOW BYTE VALUE.
010D 224000 SHLD D$ADDR ; STORE THE ADDRESS
;
; ****
; INJECT BOOT MODULE INTO CONTROLLER *
; ****
;
0110 3E01 INJECT: MVI A,DC$MBO ; REQUEST DD MEM BANK 0.
0112 D343 OUT D$PORT ; ISSUE COMMAND.
0114 01C000 LXI B,IM$SIZE ; INJECTION SIZE.
0117 EB XCHG ; D$ADDR HL TO DE.
0118 210010 LXI H,IM$ADR ; INJECTION MODULE.
011B CD5901 CALL BLOCK ; BLOCK MOVE.
;
; ****
; RESET AND START THE DISK PROCESSOR *
; ****
;
011E 3E80 MVI A,DC$BGN ; BEGIN DD PROCESSOR.
0120 D343 OUT D$PORT ; ISSUE COMMAND.
0122 E3 XTHL ; ALLOW DOUBLE D TIME.
0123 E3 XTHL ; TO START UP.
;
; ****
; WAIT FOR TASK COMPLETION *
; ****
;
0124 DB43 WAIT: IN D$PORT ; INPUT DD STATUS.
0126 E602 ANI DS$HLT ; TEST HALT* STATUS.
0128 C22401 JNZ WAIT ; WAIT TILL HALTED.

```

```

;
;*****SWITCH CONTROLLER MEMORY INTO SYSTEM BUS*****
; SWITHC CONTROLLER MEMORY INTO SYSTEM BUS      *
;*****SWITCH CONTROLLER MEMORY INTO SYSTEM BUS*****
;

012B 3E01      MVI      A,DC$MRQ      ;REQUEST MEM (BANK 0).
012D D343      OUT      D$PORT      ;ISSUE COMMAND.

;
;*****CHECK FOR BOOTSTRAP MALFUNCTION*****
; CHECK FOR BOOTSTRAP MALFUNCTION      *
;*****CHECK FOR BOOTSTRAP MALFUNCTION*****
;

012F 2A4000    LHLD     D$ADDR      ;CONTROLLER ADDRESS.
0132 117703    LXI      D,BL$DCS    ;ERROR CODE OFFSET.
0135 19        DAD      D          ;SET HL POINTER.
0136 7E        MOV      A,M        ;GET ERROR CODE.
0137 A7        ANA      A          ;TEST REGISTER.
0138 C26601    JNZ      BAD$LD    ;BAD LOAD.

;
;*****PERFORM BLOCK TRANSFER FROM DISK MEMORY*****
; PERFORM BLOCK TRANSFER FROM DISK MEMORY      *
;*****PERFORM BLOCK TRANSFER FROM DISK MEMORY*****
;

013B 2A4000    LHLD     D$ADDR      ;CONTROLLER ADDRESS.
013E 117803    LXI      D,BL$ADR    ;LOAD ADDRESS PNTR.
0141 19        DAD      D          ;SET HL POINTER.
0142 5E        MOV      E,M        ;LOW ORDER ADDR.
0143 23        INX      H          ;INCREMENT HL.
0144 56        MOV      D,M        ;HIGH ORDER ADDR.
0145 23        INX      H          ;REQUIRES BL.BSZ NEXT.
0146 4E        MOV      C,M        ;LOW ORDER LENGTH.
0147 23        INX      H          ;INCREMENT HL.
0148 46        MOV      B,M        ;HIGH ORDER LENGTH.
0149 B5        PUSH     D          ;USE AS JUMP ADDR.
014A 3E03      MVI      A,DC$MB1    ;SWITCH TO MEM BANK 1.
014C D343      OUT      D$PORT      ;ISSUE COMMAND.
014E 2A4000    LHLD     D$ADDR      ;DOUBLE D MEM ADDRESS.
0151 CD5901    CALL     BLOCK      ;MOVE BIOS MODULE.

;
;*****TRANSFER CONTROL TO OPERATING SYSTEM*****
; TRANSFER CONTROL TO OPERATING SYSTEM      *
;*****TRANSFER CONTROL TO OPERATING SYSTEM*****
;

0154 3E01      MVI      A,DC$MBO    ;SWITCH TO BANK.0
0156 D343      OUT      D$PORT      ;ISSUE COMMAND.
0158 C9        RET      .          ;GOTO BIOS COLD ENTRY.

;
;*****BLOCK MOVE SUBROUTINE (Z80 BLOCK MOVE REGISTERS)*****
; BLOCK MOVE SUBROUTINE (Z80 BLOCK MOVE REGISTERS)      *
;*****BLOCK MOVE SUBROUTINE (Z80 BLOCK MOVE REGISTERS)*****
;

0159 7E        BLOCK:   MOV      A,M        ;GET BYTE.
015A 23        BLOCK:   INX      H          ;INC POINTER.
015B EB        BLOCK:   XCHG     .          ;GET DESTINATION.
015C 77        BLOCK:   MOV      M,A        ;PUT BYTE.
015D 23        BLOCK:   INX      H          ;INC POINTER.
015E EB        BLOCK:   XCHG     .          ;GET SOURCE.
015F 0B        BLOCK:   DCX      B          ;ONE LESS TO DO.
0160 78        BLOCK:   MOV      A,B        ;GET HI COUNT.
0161 B1        BLOCK:   ORA      C          ;GET LO COUNT.
0162 C25901    BLOCK:   JNZ      BLOCK     ;FINISH LOADING.
0165 C9        RET      .         

;
;*****ERROR HAS BEEN DETECTED*****
; ERROR HAS BEEN DETECTED      *
;*****ERROR HAS BEEN DETECTED*****
;
```

```

0166 216D01    BAD$LD: LXI      H,ER$MSG      ;ERROR MESSAGE
0169 CD9701    CALL     MSG$OT      ;DISPLAY IT.
016C 76        HLT      .          ;HALT OR GOTO MONITOR.
;
016D 0D0A0A5359ER$MSG: DB      CR,LF,LF,   SYSTEM BOOT LOAD ERROR
0186 A0        DB      ' ' OR 80H    ;SET SIGN BIT.
;
; **** CONSOLE LINKAGE DEFINITIONS ****
; **** CONSOLE OUTPUT ****
;
0000 =         CNO$SP  EQU      000H      ;OUTPUT STATUS PORT.
0004 =         CNO$SB  EQU      004H      ;OUTPUT STATUS BIT.
0000 =         CNO$SI  EQU      000H      ;OUTPUT STATUS INVERT.
0001 =         CNO$DP  EQU      001H      ;OUTPUT DATA PORT.
;
000A =         LF      EQU      00AH      ;ASCII LINE FEED
000D =         CR      EQU      00DH      ;CARRAIGE RETURN
;
; **** CONSOLE INPUT ****
;
0187 F5        CNS$OT: PUSH    PSW      ;SAVE CHARACTER
0188 D800    C$WAIT: IN      CNO$SP      ;INPUT STATUS
018A EE00    XRI      CNO$SI      ;ADJUST POLARITY
018C E604    ANI      CNO$SB      ;MASK STATUS BIT
018E CA8801   JZ       C$WAIT      ;TRY AGAIN
0191 F1        POP     PSW      ;RESTORE CHARACTER
0192 E67F    ANI      07FH      ;7 BIT ASCII
0194 D301    OUT      CNO$DP      ;SEND CHARACTER
0196 C9        RET      .          ;
;
; **** DISPLAY MESSAGE SUBROUTINE ****
;
0197 F5        MSG$OT: PUSH    PSW      ;SAVE CALLER FLAGS.
0198 7E        M$REPT: MOV     A,M      ;LOAD CHARACTER.
0199 CD8701    CALL     CNS$OT      ;CONSOLE OUTPUT.
019C 7E        MOV     A,M      ;SAME CAHRACTER.
019D 23        INX     H       ;INCREMENT POINTER.
019E E680    ANI      080H      ;TEST SIGN BIT.
01A0 CA9801   JZ       M$REPT      ;ANOTHER CHARACTER.
01A3 F1        POP     PSW      ;RESTORE FLAGS.
01A4 C9        RET      .          ;
;
; **** END OF BEGIN ****
;
01A5           END     BEGIN

```

A>

